



Because Data needs context

Context is the next frontier in Data interoperability. Data needs context to be understood and usable. We already have global databases of Knowledge like Wikipedia, but we still lack the common language for this data to be used and integrated into any application.

A good example would be NFTs in the Metaverse. I can mint an NFT for a Car for a web3 Game, but unless we have a known standard, that Car cannot live in a different Game or metaverse. We need **coordination**.

Standards should be public goods. We need a way to name things in a common language. And when we say standards, we mean simple open and collaborative schemas not controlled by any organization. We need **cooperation**.

The only way we can have standards today is by going to international standardization bodies, where any standard takes months to be built. What if standards were created and evolved by the builders and users on short cycles. We need **efficiency**.

Context is a framework to coordinate standards as Public Goods

Context = Schemas + Data PODs + Universal Name Service

Context DAO

Context DAO is a set of Smart Contracts on Arweave to register and store standard-based information on Web3.

Context DAO is a coordination framework to elaborate and evolve standards as Public goods. Cooperation is not possible without a common language we all agree on. And building this common framework should be a coordination task. That's why a DAO is a perfect instrument for that.

As a **coder in web3**, I want the data in my Dapps to be interoperable, but can't find a repository with **Schemas (standards)**.

The problem

I need to create NFTs for Cars



Look for
examples or
other games



Associations/Organizations
creating Standards



Institutional
Standardization
Bodies



Hard to find/use
a Schema

Context is a Registry of Data linked to schemas and a public and collaborative repository of schemas where builders can find the standard they need and evolve it (add fields, taxonomy...).

The perfect scenario

I need to create NFTs for Cars



Find the
Schema

NFT.Car



Create/Evolve
the Schema

add NFT.Car/engine



Deploy the Data POD
for that Schema

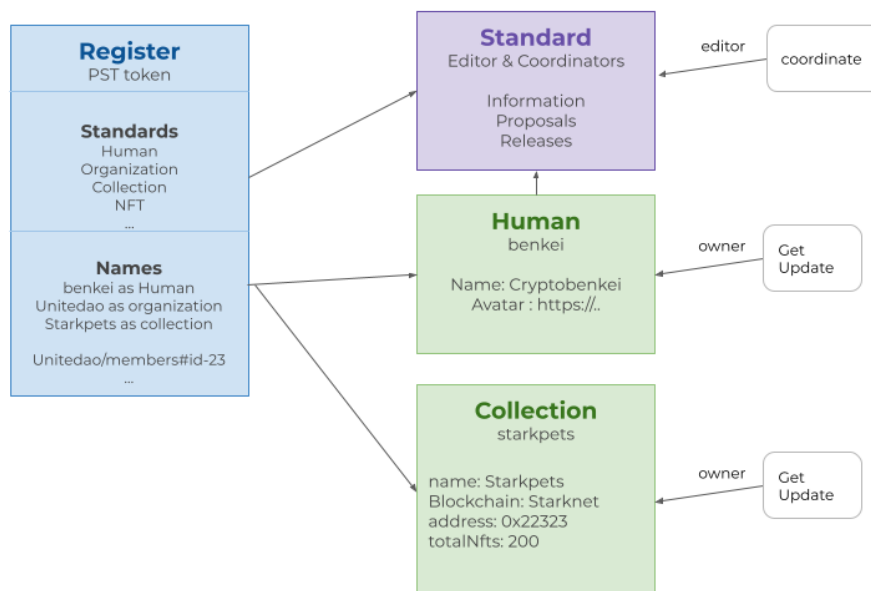


Register the
context

kitt as NFT.Car

Context is built on top of **three basic primitives**

- **Schemas** : The standards used to represent Data. Every schema is a DAO with editors and contributors.
- **Data Pods** : The data following a particular Schema. It can be edited by the owner.
- **Universal Name Service** : A Registry where schemas and data pods are referenced.



Schemas

Schemas are self contained DAOs where builders collaborate in the creation of standards. A schema is the definition of the standard, a list of typed fields following a GraphQL definition.

```
type Organization {  
  name: String!  
  url: String  
}
```

```
type NFT {  
  name: String!  
  description: String!  
  extrenalUrl: String!  
  image: String!  
}
```

```
type Collection {  
  name: String!  
  network: String!  
  address: String!  
  tokens: [NFT]!  
}
```

Schemas evolve with proposals from the contributors approved by the editors.

Solution + Arweave

Create a standard



1

Deploy a new Smart Contract for the new Schema



2

Register the name of the Standard in the Registry pointing to the address of the Schema



3

Interact with the schema : contributors proposals releases

Data Pods

Data Pods are basic units of information following a Schema. They can be edited by its owner. Data follows a standard structure (like Human) and is also a Smart Contract where some of the fields can be modified. All data is always associated to a Standard

Example: We can deploy a Organization (Context DAO), and later update the its url.

```
const unitedao: Organization = {  
  name: `ContextDAO`,  
  url: `https://ctx.xyz`  
}
```

Solution + Arweave

Register data



1

Deploy a new Smart Contract for the new Data pod.



2

Register the name of the Data Pod and its Schema, pointing to the address of the Data Pod.



3

Interact with the data

Universal Name Service

The link between the data, the standard and web3.

Example: Register johnsmith as Human to get the Address the Data for John Smith has been deployed to.

@johnsmith#city

Use Case : NFT Metadata

We have a Collection standard for NFT collections where you can update some information (we can mix mutable and immutable data) in the metadata

Tokenomics

Work In Progress : We are designing an evolutive tokenomics based on milestones. Instead of pre-minting a fixed amount of Tokens, we will work based on Goals : when x schemas are created and y data is registered a z amount of tokens will be minted and distributed between creators and users.

The token will be needed in order to register data, and every time a new

Data Pod enters the Registry, it's cost will be shared between the DAO and the contributors to the Schema that data is linked To.

Roadmap

